SIAM J. SCI. COMPUT.
Vol. 22, No. 2, pp. 535–560

© 2000 Society for Industrial and Applied Mathematics

MODELING MICROSTRUCTURE EVOLUTION USING GRADIENT-WEIGHTED MOVING FINITE ELEMENTS*

ANDREW KUPRAT†

Abstract. Microstructure evolution, where grain boundaries evolve by mean curvature motion, is modeled in three dimensions (3-D) using gradient-weighted moving finite elements (GWMFE). To do this, we modify and extend an existing 2-D GWMFE code to create a new code GRAIN3D which makes the 3-D microstructure modeling possible. The right-hand side term which drives the GWMFE motion can be viewed as surface tension forces, that is, as the negative gradient of the surface integral of a constant energy density μ on the triangular interfacial grid. Extensions to the method include equations for the motion of tetrahedra that are conformally attached to the moving piecewise linear triangular facets which represent the GWMFE discretization of the evolving grain boundaries. We present some new regularization terms which control element quality, as well as preventing element collapse in the simulation. New capabilities for changing the mesh topology are used to keep the grid edge lengths below a maximum allowable length $h_{\max}$ and to mimic actual changes in the physical topology, such as collapse and disappearance of individual grains. Validating runs are performed on some test cases that can be analytically solved, including collapse of a spherical grain and the case of columnar microstructure. In the spherical collapse case, the GWMFE method appears to have an error in the surface area collapse rate $-\frac{dA}{dt}$ which is $\mathcal{O}((\Delta\theta)^2)$, where $\Delta\theta$ is a measure of the angular resolution of the mesh. Finally, a run is presented where a true 3-D microstructure (possessing triple lines and quadruple points in the interior and triple points on the exterior boundaries) is evolved to a "2-D" columnar microstructure and finally evolved down to a single grain.

Key words. microstructure evolution, motion by mean curvature, gradient-weighted moving finite elements, unstructured tetrahedral meshes, deforming grids, changing grid topology, front-tracking

AMS subject classifications. 65M60, 51P05, 73S10, 65M50

PII. S1064827598348374

1. Introduction.

Under close examination, a sample of metal (such as copper or aluminum used in semiconductor manufacture) possesses a microstructure wherein the sample is decomposed into separate grains. The atoms in individual grains exist in a crystal lattice and the lattice orientations of adjacent grains differ. The boundary surfaces between grains are thus areas of lattice misalignment and effectively possess an excess energy per unit grain boundary area. In the simplest approximation, this excess energy density is a constant μ per unit area for all grain boundaries. When the sample is heated, the grain boundaries become mobile and grain growth takes place. The motion resulting from the desire to minimize surface energy is *mean curvature motion* where the normal velocity of a point on a grain boundary is proportional to the mean curvature at that point [19]. Thus the grain boundaries move as if they are under the influence of a driving force proportional to the mean curvature, with motion opposed by a frictional force proportional to normal velocity (as if the grain boundaries are immersed in a uniform, isotropic viscous medium).

---

*Received by the editors December 23, 1998; accepted for publication (in revised form) December 31, 1999; published electronically July 25, 2000. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights. This research was supported by the Department of Energy under contract W-7405-ENG-36.
http://www.siam.org/journals/sisc/22-2/34837.html

†Theoretical Division, Group T-1, Mailstop B-221, Los Alamos National Laboratory, Los Alamos, NM 87545 (kuprat@lanl.gov).

Approaches to mean curvature motion and/or grain growth modeling have included 2-D front-tracking models [5]; Surface Evolver, a popular energy gradient descent method [2]; level sets [23]; vertex methods [6, 9]; and gradient-weighted moving finite elements (GWMFE) [1, 3, 4, 15]. 2-*D front tracking models* have been the workhorse of grain growth modeling—they have been used in many research articles and are particularly successful in the case of thin films, where the microstructure is for the most part columnar. *Surface Evolver* is a widely used code available over the web which allows the user to interactively reduce the surface energies of arbitrary surface configurations. Constant surface energy leads to mean curvature motion, but much more general energies and motions are possible. The method uses an explicit step where mesh points are advanced along either the energy gradient descent direction or the conjugate gradient direction. *Level set methods* have the advantage of automatically resolving certain types of topological changes that occur during grain evolution such as *self-pinchoff* where a single grain divides into two or more grains. However, when three or more grains intersect at a point (*triple* or *quadruple* point) or on a curve (*triple line*), the standard level set formulation fails and a nontrivial increase in algorithmic complexity is necessary to correctly model the evolution of these points of intersection [21, 25]. *Vertex methods* consider points of maximum grain intersection (triple or quadruple points) and assume the curves between these points are straight lines. These methods may produce valid statistical results on large collections of grains but cannot hope to compute detailed grain shapes.

Moving finite elements (MFE) [1, 14, 16] is a standard Galerkin finite element method of computing the solution $u(\mathbf{x}, t)$ of a time-dependent PDE

$$u_t = F(\mathbf{x}, t, u, \nabla u, \ldots),$$

with the additional novel feature that the Galerkin formalism is used to compute domain velocities for the computational grid points in addition to determining the time derivative of u at the grid points. Because this method would frequently overconcentrate computational nodes in the steep portions of the graph of u, it was necessary to devise GWMFE which deemphasized the importance of node placement in high-gradient regions by multiplying the PDE residual by the gradient-weighting factor $\frac{1}{\sqrt{1+|\nabla u|^2}} \leq 1$. It turns out that GWMFE can be viewed as a nonweighted method when the independent variables are taken to be the nodal positions on the graph of u, the residual is taken to be the norm of errors in u_t in the direction normal to the graph of u, and the inner product integrals are taken over the surface area of the graph. In fact, GWMFE is naturally seen to be a method for evolving a surface *without* the restriction that the surface be the graph of a function u and is thus a very natural way to compute mean curvature motion. Examples of GWMFE applied to normal motion by mean curvature are given in [15, 4]; other examples of GWMFE computations involving mean curvature are given in [12, 13, 18]. Other GWMFE computations and an exposition of the method from a somewhat different viewpoint are given in the now standard book by Baines [1].

Recently, we took the 2-D version of GWMFE used in [4], which represents evolving surfaces by piecewise linear triangular elements, and incorporated it into a new code GRAIN3D which extended its capabilities so that general microstructure evolution could be simulated. Specifically, a program called LaGriT [8] was used to produce 3-D models of metallic microstructure which consisted of a domain $\Omega \subseteq \mathbb{R}^3$ partitioned into grains, each of which were composed of hundreds of tetrahedra. The triangular interfaces between the grains were extracted from the 3-D model and evolved by

GRAIN3D using the GWMFE algorithm. Additional equations of motions were devised for the "extra" vertices in the 3-D model that were interior to the grains and not on the grain boundaries. Motions were determined solely by the requirement that the tetrahedra remain well-shaped (i.e., have a low aspect ratio) as the surfaces to which they are conformally attached deform under mean curvature motion. Finally, grid topology-change software was written to maintain to some degree a roughly constant spacing between grid points as the mesh deforms and to make possible physically significant topology changes, such as the collapse and disappearance of certain grains during the course of evolution.

The new capability of supporting a system of tetrahedra that is conformally attached to the moving triangular surfaces will be crucial in the future when we use these volume elements to compute ambient quantities throughout Ω (such as temperature). This will allow more detailed simulations where the medium's resistance to grain boundary movement is not assumed uniform. (Note, for example, that grain boundary mobility in fact strongly depends on temperature [19].)

In section 2, we review GWMFE from a surface evolution point of view. We present a detailed derivation of how the GWMFE right-hand side integral involving curvature is evaluated and show that this integral can be seen as the negative gradient, with respect to the nodal position vector $\mathbf{x} = (x_i, y_i, z_i)$ of the surface integral of a constant energy per unit area μ on the discretized interfaces. In section 3, we review regularization terms necessary for the GWMFE method to work efficiently, and we introduce a right-hand regularization term involving a quality force that controls element quality as well as preventing element collapse in the simulation. We also present a regularization force that prevents excessive stretching of elements, derived from a small artificial surface energy associated with each triangular facet. In section 4, we introduce additional equations for moving tetrahedra conformally attached to the GWMFE triangular interface mesh. In section 5, we touch on grid maintenance operations necessary to keep grid edge lengths below a maximum allowable $h_{\max}$. In section 6, we describe *recoloring* operations necessary to perform important topological changes in the microstructure simulation, such as the collapse and disappearance of individual grains. In section 7, we compare numerical solutions computed by GRAIN3D to analytic solutions for the cases of spherically symmetric grain collapse and columnar microstructures. In section 8, we show a numerical example where GRAIN3D evolves a truly 3-D microstructure.

2. Mean curvature motion and GWMFE.

2.1. Review of method.

We use GWMFE [3, 4, 15] to move a multiply connected network of piecewise linear triangles for the modeling of deformation of 3-D grains. In one model of metallic grain growth [19], interface surfaces obey the simple equation

$$v_n = \mu K,$$

where v_n is the normal velocity of the interface, K is the curvature, and μ is the product of the excess energy per unit area and the mobility of the grain boundary. K is the sum of principle curvatures, i.e., twice the mean curvature. In this paper, we assume that μ is constant, i.e., it does not depend on the choice of materials on either side of the interface, nor does it depend on the orientation of the interface. We represent interfaces as parametrized surfaces:

$$\mathbf{x}(s_1, s_2) = \sum_{\text{nodes } j} \alpha_j(s_1, s_2)\mathbf{x}_j.$$

Here, (s_1, s_2) is the surface parametrization, the sum is over the N interface nodes, $\alpha_j(s_1, s_2)$ is the piecewise linear basis function (hat function) which is unity at node j and zero at all other interface nodes, and $\mathbf{x}_j = (x_j^1, x_j^2, x_j^3) \in \mathbb{R}^3$ is the vector position of node j.

We have that

$$\dot{\mathbf{x}}(s_1, s_2) = \sum_j \alpha_j(s_1, s_2)\dot{\mathbf{x}}_j,$$

is the velocity of the surface at the point $\mathbf{x}(s_1, s_2)$ (based upon linear interpolation of node velocities) and

$$v_n = \dot{\mathbf{x}}(s_1, s_2) \cdot \hat{\mathbf{n}} \quad (\hat{\mathbf{n}} \text{ is local surface normal}).$$

So

$$v_n = \sum_j (\hat{\mathbf{n}}\alpha_j) \cdot \dot{\mathbf{x}}_j. \tag{1}$$

In effect, we have that the $3N$ basis functions for v_n are $n_k\alpha_j$, where $\hat{\mathbf{n}} = (n_1, n_2, n_3)$. These basis functions are discontinuous piecewise linear, since the n_k are piecewise constant.

The GWMFE method is to minimize

$$\int (v_n - \mu K)^2 \, dS \tag{2}$$

over all possible values for the derivatives $\dot{\mathbf{x}}_i$. (The integral is over the surface area of the interfaces.) We thus obtain

$$\begin{aligned} 0 &= \frac{1}{2}\frac{\partial}{\partial \dot{x}_i^k} \int (v_n - \mu K)^2 \, dS, \quad 1 \le k \le 3, \quad 1 \le i \le N \\ &= \int (v_n - \mu K) n_k \alpha_i \, dS. \end{aligned}$$

Using (1), we obtain a system of $3N$ ODEs:

$$\left[\int \hat{\mathbf{n}}\hat{\mathbf{n}}^T \alpha_i \alpha_j \, dS\right]\dot{\mathbf{x}}_j = \int \mu K \hat{\mathbf{n}} \alpha_i \, dS,$$

or

$$\mathbf{C}(\mathbf{x})\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}), \tag{3}$$

where $\mathbf{x} = (x_1^1, x_1^2, x_1^3, x_2^1, \ldots, x_N^3)^T = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T$ is the $3N$-vector containing the x, y, and z coordinates of all N interface nodes, $\mathbf{C}(\mathbf{x})$ is the matrix of inner products of basis functions, and $\mathbf{g}(\mathbf{x})$ is the right-hand side of inner products involving surface curvature. Since $\hat{\mathbf{n}}\hat{\mathbf{n}}^T$ is a 3×3 matrix, it is clear that $\mathbf{C}(\mathbf{x})$ has a 3×3 block structure.

The ODEs are solved with an implicit second order backwards difference variable time-step ODE solver [3]. We use generalized minimal residual (GMRES) iteration [22] with block-diagonal preconditioner to solve the linear equations arising from the Newton's method. (A more complex and robust choice of iterative method and preconditioner is used in [24] to solve systems involving indefinite Jacobians that arise when GWMFE is applied to problems involving convection.) The ODEs are scaled so that the variation of the x_j^k values is $\mathcal{O}(1)$ and the truncation error η (i.e., the maximum acceptable estimated error made in the x_j^k every time step) is usually set to 10^{-3}.

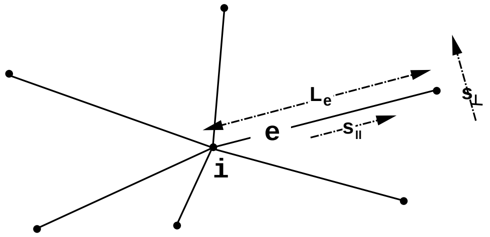


FIG. 1. *Definition of some quantities around node i, edge e.*

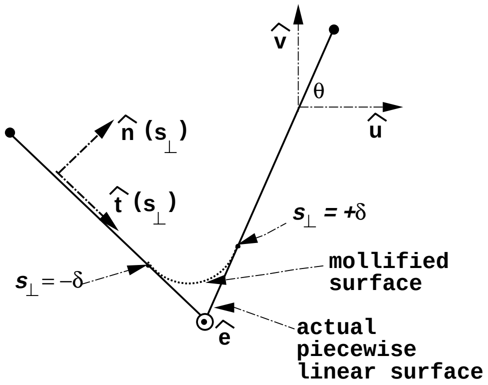


FIG. 2. *Cross-sectional view of grid. Edge e is orthogonal to page.*

2.2. Evaluation of right-hand side curvature term. The right-hand side term due to curvature,

$$\int \mu K \hat{\mathbf{n}} \alpha_i \, dS,$$

requires special consideration because on a piecewise linear manifold K is actually a distribution which is zero in the interiors of the triangles and infinite on the edges. Evaluation of this term is undertaken by mollifying (smoothing) the manifold in a small neighborhood (within a small distance δ) of the edges and then showing that $\int K \hat{\mathbf{n}} \alpha_i \, dS$ on the δ-mollified manifold tends to a limit as $\delta \to 0$ which is independent of the mollification process. This mollification interpretation is extensively explained in section 2.6 of [15], sections 2.3 and 4.10 in [3] (for 1-D), and sections 3.3 and 6.4 in [4]. For completeness we include a derivation here. Indeed, referring to Figure 1, we consider one of the edges e emanating from node i and we let $s_{||}$ be the arclength parameter running parallel to the edge and $s_\perp$ be the arclength parameter corresponding to movement on the manifold perpendicular to the edge. The length of the edge is L_e. In Figure 2, we show the intersection of the surface with a plane orthogonal to the edge e. The intersection yields a smoothed curve (due to the mollification of the surface). We assume the mollified region runs from $s_\perp = -\delta$ to $s_\perp = +\delta$. The intersection curve has tangent $\hat{\mathbf{t}}(s_\perp)$ which varies smoothly between $\hat{\mathbf{t}}(-\delta)$ and $\hat{\mathbf{t}}(+\delta)$. $\hat{\mathbf{n}}(s_\perp)$ is the normal to the surface, which smoothly varies between $\hat{\mathbf{n}}(-\delta)$ and $\hat{\mathbf{n}}(+\delta)$. We define $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$ to be unit vectors in the plane with $\hat{\mathbf{u}} \times \hat{\mathbf{v}} = \hat{\mathbf{e}}$ (the unit vector parallel to e). θ is defined as the angle that $\hat{\mathbf{t}}(s_\perp)$ makes with $\hat{\mathbf{u}}$. Now we write

$$\begin{aligned}\int \mu K \hat{\mathbf{n}} \alpha_i \, dS &= \mu \sum_{\substack{\text{edges } e \\ \text{with } i \in e}} \int_0^{L_e} \left(\int_{-\delta}^{\delta} K \hat{\mathbf{n}} \alpha_i \, ds_\perp \right) ds_{||} \\ &\approx \mu \sum_{\substack{\text{edges } e \\ \text{with } i \in e}} \left(\int_0^{L_e} \alpha_i(s_{||}) \, ds_{||} \right) \left(\int_{-\delta}^{\delta} \frac{d\theta}{ds_\perp} \hat{\mathbf{n}} \, ds_\perp \right).\end{aligned}$$

That is, the right-hand side curvature inner product at node i is the sum of contributions from edges e incident on i. In the 2δ-wide strip near edge e, $\alpha_i(s_{||}, s_\perp)$ is nearly a piecewise linear function of only $s_{||}$ which is 1 at $s_{||} = 0$ and 0 at $s_{||} = L_e$. This means that the first integral evaluates to $\frac{1}{2} L_e$. The curvature K of the mollified surface is $\frac{d\theta}{ds_\perp}$ since there is no curvature parallel to edge e in the unmollified surface (and hence no curvature in this direction for the mollified surface as well). Now

$$\begin{aligned}\int_{-\delta}^{\delta} \frac{d\theta}{ds_\perp} \hat{\mathbf{n}} \, ds_\perp &= \int_{\theta(-\delta)}^{\theta(\delta)} \hat{\mathbf{n}} \, d\theta \\ &= \int_{\theta(-\delta)}^{\theta(\delta)} (-\sin\theta \hat{\mathbf{u}} + \cos\theta \hat{\mathbf{v}}) \, d\theta \\ &= \cos\theta \hat{\mathbf{u}} + \sin\theta \hat{\mathbf{v}} \Big|_{\theta(-\delta)}^{\theta(\delta)} \\ &= \hat{\mathbf{t}}(\delta) - \hat{\mathbf{t}}(-\delta).\end{aligned}$$

So we obtain as $\delta \to 0$ that

$$\int \mu K \hat{\mathbf{n}} \alpha_i \, dS = \mu \sum_{\substack{\text{edges } e \\ \text{with } i \in e}} \frac{1}{2} L_e (\hat{\mathbf{t}}_e^{(1)} + \hat{\mathbf{t}}_e^{(2)}),$$

where $\hat{\mathbf{t}}_e^{(1)}$, $\hat{\mathbf{t}}_e^{(2)}$ are unit normals in the piecewise linear surface which are both orthogonal to e. $\hat{\mathbf{t}}_e^{(1)}$ points into one triangle sharing e, and $\hat{\mathbf{t}}_e^{(2)}$ points into the other triangle sharing e. Now we can rewrite this sum as a sum over the triangles incident on i:

$$\int \mu K \hat{\mathbf{n}} \alpha_i \, dS = \mu \sum_{\substack{\text{triangles } k \\ \text{with } i \in k}} \frac{1}{2} \big(L_k^{(1)} \hat{\mathbf{t}}_k^{(1)} + L_k^{(2)} \hat{\mathbf{t}}_k^{(2)} \big),$$

where for each triangle k, $L_k^{(1)}$ is the length of one of the edges bordering k that contains i, $\hat{\mathbf{t}}_k^{(1)}$ is the inward normal to that edge, and $L_k^{(2)}$, $\hat{\mathbf{t}}_k^{(2)}$ are the corresponding quantities for the other edge. (See Figure 3.)

Interestingly, since $\sum_{j=1}^{3} L_k^{(j)} \hat{\mathbf{t}}_k^{(j)} = 0$ (again, see Figure 3 for the definition of $L_k^{(3)}$, $\hat{\mathbf{t}}_k^{(3)}$), we have

$$\begin{aligned}\int \mu K \hat{\mathbf{n}} \alpha_i \, dS &= -\mu \sum_{\substack{\text{triangles } k \\ \text{with } i \in k}} \frac{1}{2} L_k^{(3)} \hat{\mathbf{t}}_k^{(3)} \\ &= -\nabla_{\mathbf{x}_i} \Bigg(\sum_{\substack{\text{triangles } k \\ \text{with } i \in k}} \mu A_k \Bigg),\end{aligned}$$

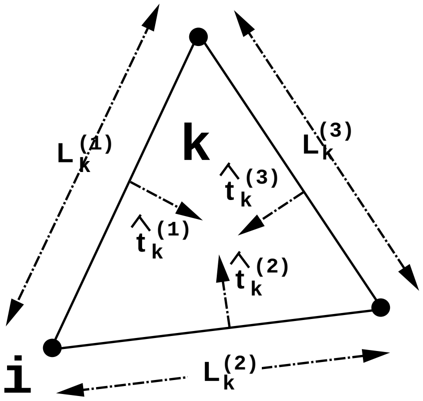


FIG. 3. *Definition of quantities around node i, triangle k.*

where A_k is the area of triangle k. If we attribute an energy of μ per unit area to the surface, then this becomes

$$\int \mu K \hat{\mathbf{n}} \alpha_i \, dS = -\nabla_{\mathbf{x}_i} E, \tag{4}$$

where

$$E = \sum_{\text{triangles } k} \mu A_k \tag{5}$$

is the total energy of the surface. This shows that the node-concentrated right-hand side force on the ith node is derivable from the negative gradient, with respect to the node position variables $\mathbf{x}_i = (x_i, y_i, z_i)$ of the surface integral of an assumed constant surface energy density of μ in our GWMFE manifold. This mirrors the physical model where mean curvature motion was derived under the assumption that the mechanism of grain growth is minimization of excess surface energy proportional to the surface integral of μ. This negative gradient force on the ith node is exactly that which would be given by a "surface tension" of magnitude μ in the planar triangular cells of our GWMFE manifold, as pointed out in [15, 4].

We note here that the form (4)–(5) for the right-hand side PDE driving terms makes conceivable computations where $\mu = \mu(\hat{\mathbf{n}})$, that is, where μ depends on the direction of the normal at the surface. In this case, one need only loop through triangles in the mesh and evaluate the quantities (4)–(5) using the orientation-dependent μ. If μ depends strongly on $\hat{\mathbf{n}}$, faceting of the surface can occur [7].

3. Regularization.

In [15] it is shown how (3) represents a balance of forces on the moving GWMFE surface. The right-hand side represents driving forces from the PDE, and the left-hand side represents viscosity forces that resist these driving forces; all these forces are taken to be concentrated at the nodes. In practice, certain grid configurations can cause $\mathbf{C}(\mathbf{x})$ to be nearly singular, so that there exist essentially undamped node motions that lead to chaotic grid behavior and collapse of the time step. To alleviate this, Carlson and Miller [4] added a term (a left-hand side regularizing force) that was of the form $\mathbf{C}_1(\mathbf{x})\dot{\mathbf{x}}$ so that the total matrix $(\mathbf{C} + \mathbf{C}_1)(\mathbf{x})$ was positive definite.

In addition certain grid behaviors are possible, such as the collapse to zero of the inscribed radius of a very thin or large aspect ratio triangle, which has no physical significance (i.e., which does not affect the shape of the surface) but which causes a catastrophic loss of time step and run termination. To alleviate these problems, small regularizing forces are added to the right-hand side of (3). The form of these forces differs from those used in [4].

3.1. Left-hand side grid viscosity forces. Suppose a node i and its neighboring nodes are all nearly coplanar. More precisely, suppose that there exists a plane P (spanned by $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$) such that $\mathbf{x}_i$ and each member of $\{\mathbf{x}_j \mid \text{nodes } i \text{ and } j \text{ share an edge}\}$ have normal distance $\leq \eta$ to this plane. Then, since the local truncation error tolerance is η, computationally this situation is indistinguishable from the case where $\mathbf{x}_i$ and all the $\mathbf{x}_j$ exactly lie in the same plane. In this case, it is apparent that one could alter $\dot{\mathbf{x}}_i$ (or $\mathbf{x}_i$) by a plane vector $a\hat{\mathbf{u}} + b\hat{\mathbf{v}}$ without altering the normal velocity (or the shape) of the surface in the neighborhood of node i. Since (3) is derived from (2), which is a minimization that is unaffected by these additional node motions for node i, we conclude that $\mathbf{C}(\mathbf{x})$ nearly annihilates the 2-D subspace corresponding to the in-plane motions of node i. If this degeneracy or near-degeneracy is not remedied, the motion of node i will not be smooth, if it is defined at all, and the time step will decrease greatly. (This describes a *planar*-type degeneracy at node i; there are also possible *crease*-type geometrical degeneracies in which the manifold has a straight crease through node i, as in the straight ridge of a roof. See section 3.1 of [15] or section 2.4 of [4] for a further discussion and for a derivation of the following grid viscosity forces for GWMFE. Also see Chapters 6 and 7 of [1].)

To remedy such degeneracies, we add the small grid viscosity force

$$\epsilon_1 \left[\int \mathbf{I}_3 (\nabla_{\mathbf{s}} \alpha_i \cdot \nabla_{\mathbf{s}} \alpha_j) \, dS \right] \dot{\mathbf{x}}_j \equiv \mathbf{C}_1(\mathbf{x}) \dot{\mathbf{x}} \tag{6}$$

to the left-hand side of (3). Here $\mathbf{I}_3$ represents the 3×3 identity matrix, which indicates that grid viscosity is isotropic in that it treats independently and identically the x, y, and z components of the velocities $\dot{\mathbf{x}}_j$. ϵ_1 is a small number, and the gradient $\nabla_{\mathbf{s}}$ is with respect to the tangential coordinate system of the surface. We now show (6) is equal to

$$\epsilon_1 \nabla_{\dot{\mathbf{x}}_i} \frac{1}{2} \int ||\nabla_{\mathbf{s}} \dot{\mathbf{x}}||^2 \, dS. \tag{7}$$

Here $\nabla_{\dot{\mathbf{x}}_\mathbf{i}}$ is the gradient with respect to the node velocity $\dot{\mathbf{x}}_i$ and $\dot{\mathbf{x}} = \sum \dot{\mathbf{x}}_j \alpha_j(\mathbf{s})$ is interpolated grid velocity. $\nabla_{\mathbf{s}} \dot{\mathbf{x}}$ is the gradient with respect to surface measure of interpolated grid velocity and, hence, it is a 3×2 matrix. $||\nabla_{\mathbf{s}} \dot{\mathbf{x}}||^2$ is the sum of squares of the six components (i.e., the Froebenius norm). To show (7), we have that

$$\begin{aligned}
\nabla_{\dot{\mathbf{x}}_i} \frac{1}{2} \int ||\nabla_{\mathbf{s}} \dot{\mathbf{x}}||^2 \, dS &= \nabla_{\dot{\mathbf{x}}_i} \frac{1}{2} \int \Big|\Big|\nabla_{\mathbf{s}} \sum_k \dot{\mathbf{x}}_k \alpha_k \Big|\Big|^2 \, dS \\
&= \nabla_{\dot{\mathbf{x}}_i} \frac{1}{2} \int \sum_k \sum_j (\dot{\mathbf{x}}_k \cdot \dot{\mathbf{x}}_j)(\nabla_{\mathbf{s}} \alpha_k \cdot \nabla_{\mathbf{s}} \alpha_j) \, dS \\
&= \int \sum_j \dot{\mathbf{x}}_j (\nabla_{\mathbf{s}} \alpha_i \cdot \nabla_{\mathbf{s}} \alpha_j) \, dS \\
&= \sum_j \left[\int \mathbf{I}_3 (\nabla_{\mathbf{s}} \alpha_i \cdot \nabla_{\mathbf{s}} \alpha_j) \, dS \right] \dot{\mathbf{x}}_j.
\end{aligned}$$

Thus, our regularization force is, for each node, the velocity gradient of a velocity potential and the effect of this force is to reduce $\int ||\nabla_{\mathbf{s}}\dot{\mathbf{x}}||^2 \; dS$, a measure of the nonuniformity of the interpolated velocity field.

Using arguments in [3], ϵ_1 is chosen in the range

$$.25\eta^2 \leq \epsilon_1 \leq 25\eta^2, \tag{8}$$

so that the effect of the regularization term dominates only when $\mathbf{x}_i$ and the neighboring $\mathbf{x}_j$ are within η of an exact plane. If the graph is not nearly planar in this sense, the left-hand side will be dominated by $\mathbf{C}(\mathbf{x})\dot{\mathbf{x}}$, which represents viscous node resistance arising from the minimization principle (2).

3.2. Right-hand side triangle quality force. Minimization of (2) does not prevent the collapse of triangles in the moving grid; if triangle collapse (i.e., shrinking to zero of inscribed radius) is allowed to occur, the numerical run will have to be terminated. To prevent triangle collapse, a right-hand side regularizing quality force is added of the form

$$-\epsilon_2 \nabla_{\mathbf{x}_i} Q^{\text{tri}}, \tag{9}$$

where Q^{tri} is a dimensionless quality energy:

$$Q^{\text{tri}} = \sum_{\text{triangles k}} Q_k^{\text{tri}} = \sum_{\text{triangles k}} \left[\frac{(L_k^{(1)})^2 + (L_k^{(2)})^2 + (L_k^{(3)})^2}{A_k} \right]^2, \tag{10}$$

where the $L_k^{(i)}$ are edge lengths for triangle k and A_k is the area. Q_k^{tri} has a minimal value of $Q_{\text{eql}}^{\text{tri}} = 48$ for an equilateral triangle and gets larger as the aspect ratio of the triangle gets larger. That is, poorly shaped triangles have poor quality and a large quality energy. Q_k^{tri} is dimensionless and is thus independent of the scale of the triangle k. However, the force $-\nabla_{\mathbf{x}_i} Q_k^{\text{tri}}$ is not scale invariant and its magnitude increases as the scale is decreased. We can thus choose ϵ_2 to be equal to a value such that $-\nabla_{\mathbf{x}_i} Q_k^{\text{tri}}$ dominates for small triangles with shortest altitude $\leq \eta$. This will prevent triangle collapse. Indeed, for triangle k (assuming the nomenclature of Figure 3), we have that

$$\begin{aligned}
-\nabla_{\mathbf{x}_i}\epsilon_2 Q_k^{\text{tri}} &= -\epsilon_2 \nabla_{\mathbf{x}_i} \left(\frac{\sum_{j=1}^3 (L_k^{(j)})^2}{A_k} \right)^2 \\
&= -2\epsilon_2 \frac{\sum_{j=1}^3 (L_k^{(j)})^2}{A_k} \left(\frac{2\sum_{j=1}^3 L_k^{(j)} \nabla_{\mathbf{x}_i} L_k^{(j)}}{A_k} - \frac{\sum_{j=1}^3 (L_k^{(j)})^2}{A_k} \frac{\nabla_{\mathbf{x}_i} A_k}{A_k} \right) \\
&= -2\epsilon_2 Q_k^{\text{tri}} \left(\frac{2(L_k^{(1)} \nabla_{\mathbf{x}_i} L_k^{(1)} + L_k^{(2)} \nabla_{\mathbf{x}_i} L_k^{(2)})}{\sum_{j=1}^3 (L_k^{(j)})^2} - \frac{\nabla_{\mathbf{x}_i} A_k}{A_k} \right) \\
&= 2\epsilon_2 Q_k^{\text{tri}} \left(-\frac{2L_k^{(1)}}{||L_k||^2} \nabla_{\mathbf{x}_i} L_k^{(1)} - \frac{2L_k^{(2)}}{||L_k||^2} \nabla_{\mathbf{x}_i} L_k^{(2)} + \frac{\hat{\mathbf{t}}_k^{(3)}}{h_k^{(3)}} \right),
\end{aligned}$$

where $h_k^{(3)}$ is the altitude of the vertex $\mathbf{x}_i$ above the opposite edge in triangle k and $||L_k||^2$ is the square of the l_2 norm of the edge lengths. Since $-\nabla_{\mathbf{x}_i} L_k^{(1)}$ and $-\nabla_{\mathbf{x}_i} L_k^{(2)}$ are unit vectors directed along the edges in triangle k that originate at $\mathbf{x}_i$, we have that the first two terms represent tension forces of magnitudes $4\epsilon_2 Q_k^{\text{tri}} \frac{L_k^{(1)}}{||L_k||^2}$ and $4\epsilon_2 Q_k^{\text{tri}} \frac{L_k^{(2)}}{||L_k||^2}$, respectively, which act along edges that attract node $\mathbf{x}_i$ to its neighboring nodes in triangle k. Since $\hat{\mathbf{t}}_k^{(3)}$ is a unit vector pointing normally towards node $\mathbf{x}_i$ from the edge opposite $\mathbf{x}_i$, the third term represents a repulsive force of magnitude $\frac{2\epsilon_2 Q_k^{\text{tri}}}{h_k^{(3)}}$ that prevents triangle collapse. The quality force thus causes a node to be attracted along edges and repelled along altitudes. It is zero only for the case of an equilateral triangle. If the triangle is nonequilateral, there will be a node with a shortest altitude that will feel a net repulsion from the opposite edge. If the triangle shape becomes very distorted, the node will feel a repulsive force that is arbitrarily larger than the attractive forces directed along the edges.

Assuming the case of distorted, nonequilateral triangles with shortest altitude at $\mathbf{x}_i$, we wish the dominant repulsive force be comparable to the "physically justified" driving force arising from the PDE, $-\nabla_{\mathbf{x}_i} \int \mu \, dS$, when the shortest altitude of the triangle is of order η. Thus, referring to (5), we set

$$\begin{aligned} \frac{2\epsilon_2 Q_k^{\text{tri}}}{h_k^{(3)}} &= || - \nabla_{\mathbf{x}_i} \mu A_k || \\ &= \frac{\mu L_k^{(3)}}{2}, \end{aligned}$$

so that

$$\epsilon_2 = \frac{\mu L_k^{(3)} h_k^{(3)}}{4 Q_k^{\text{tri}}}.$$

Since we are assuming $Q_k^{\text{tri}} > Q_{\text{eql}}^{\text{tri}}$ and $L_k^{(3)} > h_k^{(3)} = \mathcal{O}(\eta)$, to obtain an order of magnitude estimate for ϵ_2 we insert the trial values $Q_k^{\text{tri}} = 10\ Q_{\text{eql}}^{\text{tri}}$ and $h_k^{(3)} = \eta$ (which implies that roughly $L_k^{(3)} = 6\ \eta$). This leads to

$$\epsilon_2 = 3 \times 10^{-3}\ \mu \eta^2. \tag{11}$$

Certainly with $Q_k^{\text{tri}} = 10\ Q_{\text{eql}}^{\text{tri}}$ we have that the repulsive force along the triangle altitude dominates, as assumed. If $Q_k^{\text{tri}} > 10\ Q_{\text{eql}}^{\text{tri}}$, for triangles with shortest altitude of order η (or larger than η for sufficiently distorted triangles) the "quality force" is stronger than the physical tension force. This means that large aspect ratio triangles are prevented from forming or are "locked up," prevented from attaining an even worse shape. In practice, these needle-shaped triangles make up a small amount of the surface area of the computational interface surfaces and do not significantly affect the accuracy of the computed solutions. If $Q_k^{\text{tri}} < 10\ Q_{\text{eql}}^{\text{tri}}$, the quality force is weaker and for Q_k^{tri} approaching $Q_{\text{eql}}^{\text{tri}}$ we have that the attractive forces along edges begin to cancel the repulsive force until at $Q_k^{\text{tri}} = Q_{\text{eql}}^{\text{tri}}$ there is no force whatsoever. Thus the force (9)–(10) *does* allow for the collapse of triangles *if they remain perfectly equilateral* during collapse. However, in practice, there is always some asymmetry in the computational runs, and so (9) with choice (11) is indeed effective at preventing triangle collapse. In fact, ϵ_2 is chosen to be somewhat less than the estimate (11) since we have found that the choice of ϵ_2 (and of the other regularization coefficients ϵ_i in this paper) is not critical and we can err on the side of introducing *less* artificial regularization forces.

3.3. Right-hand side perimeter-dependent surface energy. The opposite of grid collapse is excessive stretching of triangular elements. If edges exceed a certain maximum acceptable edge length $h_{\max}$, then the GWMFE method is running on too coarse a discretization, degrading accuracy of the results. Even if the initial grid has all edges $\leq h_{\max}$ in length without explicit intervention, it is quite possible that some edges exceed $h_{\max}$ in length after some grid evolution. To discourage excessive grid stretching, we add a right-hand side surface energy that is proportional to the perimeter of the triangles:

$$\begin{aligned} -\nabla_{\mathbf{x}_i} \epsilon_3 E^{\text{perim}} &= -\nabla_{\mathbf{x}_i} \left(\epsilon_3 \sum_{\text{triangles k}} E_k^{\text{perim}} \right) \\ &= -\nabla_{\mathbf{x}_i} \left(\epsilon_3 \sum_{\text{triangles k}} (L_k^{(1)} + L_k^{(2)} + L_k^{(3)}) A_k \right). \end{aligned} \tag{12}$$

This corresponds to an artificial energy density $\mu_3 \equiv \epsilon_3 p_k$, where $p_k = L_k^{(1)} + L_k^{(2)} + L_k^{(3)}$ is the triangle perimeter. This force grows in strength as the perimeter gets large, and thus excessive stretching of individual elements will be avoided. Since it is highly likely that most edges will be near $h_{\max}$ in length, the artificial energy density will constantly be affecting the solution, introducing errors throughout the grid. (This is as opposed to the triangle quality force which will only affect the solution near "hot spots" where the grid has "thin" elements nearing collapse.) Errors in surface velocities due to (12) will be roughly $\mathcal{O}(\mu_3/\mu)$ throughout the grid, so we want this quantity to be less than some tolerance. On our $\mathcal{O}(1)$ grid, η represents a tolerance on fractional error in node position. It is reasonable to also take this as fractional error in surface velocity, and so using $\mu_3 \approx \epsilon_3(3h_{\max})$, we obtain

$$\epsilon_3 \lesssim \frac{\eta\mu}{3h_{\max}}. \tag{13}$$

4. Enlargement of system to move noninterface nodes. System (3) is integrated using a second-order implicit variable time step ODE solver [3]. However, it gives velocities of the interface nodes only ($\dot{\mathbf{x}} = (\dot{x}_j^k)_{,1\leq l\leq N}^{,1\leq k\leq 3}$), and so the system must be enlarged to include velocities for M interior nodes that are not part of the interface. That is, we extend $\mathbf{x}$ to

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_{interface} \\ \mathbf{x}_{interior} \end{pmatrix},$$

where $\mathbf{x}_{interface} = (x_j^k)_{,1\leq l\leq N}^{,1\leq k\leq 3}$, and $\mathbf{x}_{interior} = (x_j^k)_{,N+1\leq l\leq N+M}^{,1\leq k\leq 3}$. With this extension, we enlarge system (3) to be order $N + M$.

Since interface physics only tells us how to evolve the N interface nodes, we must "artificially" construct the extra elements in the enlarged $\mathbf{C}(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$ to allow for orderly (tetrahedra orientation preserving) evolution of the mesh. That is, given a physically meaningful method of evolving the triangular interfaces, we are free to develop auxiliary equations for moving the tetrahedra (some of which are conformally attached to the triangular interfaces) with the only requirements being that these equations lead to efficient solution of the system (3) and that they maintain positive orientation of tetrahedra.

A natural choice for the additional equations is to generalize the left-hand side regularization force (6) to viscously damp volume grid motions, and to generalize the right-hand side quality force (9)–(10) to seek well-shaped tetrahedra.

4.1. Volume grid viscosity force. To the left-hand side of (3), we add the following contribution:

$$\epsilon_4 \left[\int \mathbf{I}_3 (\nabla\tilde{\alpha}_i \cdot \nabla\tilde{\alpha}_j) \, dV \right] \dot{\mathbf{x}}_j \equiv \mathbf{C}_4(\mathbf{x})\dot{\mathbf{x}}. \tag{14}$$

Here, $\nabla\tilde{\alpha}_j = \nabla\tilde{\alpha}_j(\mathbf{x})$ is a piecewise linear hat function defined for $\mathbf{x} \in \mathbb{R}^3$. $\tilde{\alpha}_j$ is 1 on the jth node in the tetrahedral mesh and zero on all other nodes. $\nabla\tilde{\alpha}_j$ is the gradient of $\tilde{\alpha}_j$ over the tetrahedra (i.e., $\nabla = \nabla_{\mathbf{x}}$, where $\mathbf{x} \in \mathbb{R}^3$.) The integral is taken over the whole volume domain $\Omega \subseteq \mathbb{R}^3$ which has been partitioned into tetrahedra. This is a generalization of the surface grid viscosity term (6). Similar to (6)–(7), (14) is easily seen to be equal to

$$\epsilon_4 \nabla_{\dot{\mathbf{x}}_i} \frac{1}{2} \int ||\nabla\dot{\mathbf{x}}||^2 \, dV, \tag{15}$$

where $\dot{\mathbf{x}} = \sum_j \dot{\mathbf{x}}_j \tilde{\alpha}_j(\mathbf{x})$ is interpolated volume mesh velocity. As in the case of surface grid viscosity, the effect of (14) is to reduce nonuniformities in volume grid velocity. So, for example, if a collection of noninterface nodes were surrounded by a set of interface nodes all moving at constant velocity $\mathbf{a}$, the solution of the system

$$\begin{aligned} \mathbf{C}_4 \, \dot{\mathbf{x}} &= 0, \\ \dot{\mathbf{x}}_i &= \mathbf{a} \quad \text{at interface nodes} \end{aligned}$$

would be $\dot{\mathbf{x}} = \mathbf{a}$ at noninterface nodes as well. This is clear, since in this case $\int ||\nabla\dot{\mathbf{x}}||^2 \, dV$ is zero and is thus minimized.

To choose a suitable value for ϵ_4, we compare (6) and (14) and arrange for these terms to be roughly of the same order. Suppose node i is on the interfacial triangular network. We choose ϵ_4 so that the diagonal elements of the matrices $\mathbf{C}_1$ and $\mathbf{C}_4$ are roughly the same. That is,

$$\epsilon_4 \int \nabla\tilde{\alpha}_i \cdot \nabla\tilde{\alpha}_i \, dV \approx \epsilon_1 \int \nabla_{\mathbf{s}}\alpha_i \cdot \nabla_{\mathbf{s}}\alpha_i \, dS.$$

If we assume that the altitudes of the volume elements sharing node i are about the same magnitude as the altitudes of the surface elements sharing node i, then the integrands of the two integrals are roughly the same. Thus we should have

$$\begin{aligned} \epsilon_4 &\approx \epsilon_1 \int \nabla_{\mathbf{s}}\alpha_i \cdot \nabla_{\mathbf{s}}\alpha_i \, dS \Big/ \int \nabla\tilde{\alpha}_i \cdot \nabla\tilde{\alpha}_i \, dV \\ &\approx \epsilon_1 \int_{|\mathbf{x}-\mathbf{x}_i| \le h} dS \Big/ \int_{|\mathbf{x}-\mathbf{x}_i| \le h} dV \\ &= \frac{3}{4}\frac{1}{h}\epsilon_1, \end{aligned}$$

where h is the length of a typical edge in the mesh near node i. We approximate h by $h_{\max}$, since in practice many edges are at (or near) the longest allowed edge length in the computation. Thus, using (8), we arrive at the guideline,

$$.2\eta^2/h_{\max} \le \epsilon_4 \le 20\eta^2/h_{\max}. \tag{16}$$

4.2. Volume quality force. Similar to section 3.2, to prevent tetrahedral collapse and inversion as the mesh evolves, to the right-hand side of (3) we add the tetrahedral quality force

$$-\epsilon_5 \nabla_{\mathbf{x}_i} Q^{\text{tet}}, \tag{17}$$

where Q^{tet} is the dimensionless quality energy

$$Q^{\text{tet}} = \sum_{\text{tets } p} Q_p^{\text{tet}} = \sum_{\text{tets } p} \frac{\sum_{n=1}^{6} (L_p^{(n)})^2 \sum_{n=1}^{4} (A_p^{(n)})^2}{V_p^2}, \tag{18}$$

where the $L_p^{(n)}$ are the edge lengths, the $A_p^{(n)}$ are the face areas, and V_p is the volume of tetrahedron p. Q_p is a dimensionless quality measure which has the minimal value $Q_{\text{eql}}^{\text{tet}} = 324$ if p is a regular (equilateral) tetrahedron but approaches infinity if p has a worsening aspect ratio. We note that Q_k^{tri} (10) and Q_p^{tet} are both equivalent to $||L||^2||\frac{1}{H}||^2$, the square of the l_2 norm of the edge lengths of the element multiplied by the square of the l_2 norm of the inverse altitudes of the element. They are thus dimensionless element quality measures which are extremely smooth (i.e., require no square root evaluations).

The argument for setting ϵ_5 is similar to that for setting ϵ_2. Suppose one face k of a nearly collapsed tetrahedron p is on an interface. Suppose node i is on this face. We choose ϵ_5 so the force on this node due to the quality force begins to dominate for small nonregular tetrahedra with shortest altitude $\leq \eta$. Analogous to the triangle quality force of section 3.2, node i is attracted along the edges of tetrahedron p emanating from the node (terms involving $\nabla_{\mathbf{x}_i} L_p^{(n)}$), node i is attracted along the altitudes of the faces emanating from the node (terms involving $\nabla_{\mathbf{x}_i} A_p^{(n)}$), and node i is repelled along the altitude from the face opposite the node (the term involving $\nabla_{\mathbf{x}_i} V_p$). The quality force is zero only for the case of a regular tetrahedron. If the tetrahedron is nonregular, there will be a node with a shortest altitude that will feel a net repulsion from the opposite face. If the tetrahedron shape becomes very distorted, the node will feel a repulsive force that is arbitrarily larger than the attractive forces directed along the edges and faces. So assuming that the shortest altitude (and hence greatest quality force) occurs at $\mathbf{x}_i$, we have

$$\begin{aligned}
-\epsilon_5 \nabla_{\mathbf{x}_i} Q_p^{\text{tet}} &= -\epsilon_5 \nabla_{\mathbf{x}_i} \frac{\sum_{n=1}^{6} (L_p^{(n)})^2 \sum_{n=1}^{4} (A_p^{(n)})^2}{V_p^2} \\
&\approx 2\epsilon_5 \frac{\sum_{n=1}^{6} (L_p^{(n)})^2 \sum_{n=1}^{4} (A_p^{(n)})^2}{V_p^2} \frac{\nabla_{\mathbf{x}_i} V_p}{V_p} \\
&= \frac{2\epsilon_5 Q_p^{\text{tet}}}{h_p^i} \hat{\mathbf{t}}_p^i,
\end{aligned}$$

where h_p^i is the altitude of point $\mathbf{x}_i$ above the opposite face in tetrahedron p, and $\hat{\mathbf{t}}_p^i$ is a unit vector pointing along the altitude (towards $\mathbf{x}_i$). As in section 3.2, we wish the magnitude of this force to be equal to the magnitude of the physical force $-\nabla_{\mathbf{x}_i} \mu A_k$ due to surface tension over the interface triangle k of tetrahedron p. Again using the nomenclature of Figure 3, we have

$$\frac{2\epsilon_5 Q_p^{\mathrm{tet}}}{h_p^i} = || - \nabla_{\mathbf{x}_i} \mu A_k|| \\ = \frac{\mu L_k^{(3)}}{2}.$$

So

$$\epsilon_5 = \frac{\mu L_k^{(3)} h_p^i}{4 Q_p^{\mathrm{tet}}}.$$

To obtain an order of magnitude estimate for ϵ_5, we insert the trial values $Q_p^{\mathrm{tet}} = 10\ Q_{\mathrm{eql}}^{\mathrm{tet}}$ and $h_p^i = \eta$, which implies that roughly $L_k^{(3)} = 5\ \eta$ (see below). This leads to

$$\epsilon_5 \approx 4 \times 10^{-4} \mu \eta^2. \tag{19}$$

We justify the assignment $L_k^{(3)}/h_p^i = 5$ when $Q_p^{\mathrm{tet}} = 10\ Q_{\mathrm{eql}}^{\mathrm{tet}}$ as follows. We assume face k is roughly equilateral (the force in section 3.2 presuming to have prevented poor aspect ratio triangles on the interface). So say $L_k^{(1)} = L_k^{(2)} = L_k^{(3)} = L$. But we are assuming tetrahedron p has a short altitude h_p^i at node $\mathbf{x}_i$ pointing away from interface triangle k into the volume. Given the classification of nearly degenerate tetrahedra in [11, p. 286], tetrahedron p either has a single short edge at $\mathbf{x}_i$ or no short edges at all. In both cases, a rough calculation shows $L/h_p^i \approx 5$ when $Q_p^{\mathrm{tet}} = 10\ Q_{\mathrm{eql}}^{\mathrm{tet}}$. The remarks at the end of section 3.2 (which discuss the cases $Q < 10$ and $Q > 10$ for triangles) are analogously true here for tetrahedra. Again, these are only order of magnitude estimates which are adequate for setting regularization force coefficients.

4.3. Tetrahedron lock-up. The grid forces acting on tetrahedra move the grid by (1) acting to minimize nonuniformities in grid velocity, and (2) acting to continually improve grid element aspect ratios. The artificial grid forces have the effect of necessarily overriding physically justified node movement when it is necessary to prevent inversion of tetrahedra. For instance, if a tetrahedron p has all 4 nodes on an interface, the motion given by (3) might cause the tetrahedron to invert, especially if the interface changes its sense of curvature. By adding the quality force (17) and (18), the tetrahedron will lock up when the shortest altitude is close to η and will be prevented from inverting. The effect of this on the simulation is acceptable with regard to accuracy since the lock up of a small number of tetrahedra simply removes a small fraction of numerical degrees of freedom from the simulation. The effect is further reduced if the locked up tetrahedra are effectively removed by merge and swap operations mentioned in the next section.

4.4. Equation summary. After adding regularization and tetrahedral dynamics terms to (3), the complete system of ODEs that we solve in our implementation of GWMFE is

$$\left[\int \left(\hat{\mathbf{n}}\hat{\mathbf{n}}^T \alpha_i \alpha_j + \mathbf{I}_3 \epsilon_1 (\nabla_{\mathbf{s}} \alpha_i \cdot \nabla_{\mathbf{s}} \alpha_j) \right) dS \ + \ \int \mathbf{I}_3 \epsilon_4 (\nabla \tilde{\alpha}_i \cdot \nabla \tilde{\alpha}_j)\ dV \right] \dot{\mathbf{x}}_j \\ = \int \mu K \hat{\mathbf{n}} \alpha_i \ dS \ - \ \nabla_{\mathbf{x}_i} (\epsilon_2 Q^{\mathrm{tri}} + \epsilon_3 E^{\mathrm{perim}} + \epsilon_5 Q^{\mathrm{tet}}) \ , \quad 1 \le i \le N + M. \tag{20}$$

Here the system has been written as $N + M$ 3-vector equations, one associated with each of the N interface nodes and M interior nodes. The left-hand side is an implied sum over all the nodes j in the mesh. The "artificial" terms involving $\epsilon_1, \dots, \epsilon_5$ and guidelines for setting $\epsilon_1, \dots, \epsilon_5$ are explained in (6), (8), (9)–(11), (12), (13), (14), (16), (17)–(19). For the left-hand side, if either i or j is not an interface node, then the surface integral is not defined and does not contribute to the left-hand side. On the right-hand side of (20), if i is not an interface node, then only the term involving Q^{tet} is defined and contributes to the right-hand side.

5. Grid maintenance operations.

As the surfaces move and grains deform, mesh maintenance and mesh optimization tools are used to ensure good element quality and to ensure grid edges do not stretch beyond the allowable maximum length $h_{\max}$. Primitive grid operations provided by periodic calls to LaGriT, the Los Alamos Grid Toolbox [8], provide a basis for mesh maintenance and optimization. The *merge* primitive accepts as input lists of pairs of neighboring nodes: merge candidate nodes and survivor nodes. If the merge is completed, only one of the pair survives and the mesh connectivity is repaired to reflect that one node has been deleted. Before the merge takes place, LaGriT verifies that the merge will not cause tetrahedra to become inverted and that the node types and surface constraints of the survivor and merged nodes will lead to a legal merge. The *refine* primitive used in these simulations adds nodes at the midpoints of selected edges. LaGriT sets the node type and surface constraints of the added nodes by determining if the added node is in the interior, on a material interface, or on an exterior boundary. The grid connectivity is repaired to include the new elements created by connecting the new node to the other vertices of the elements which contained the refined edge. Depending on what is desired by the user, the *recon* primitive swaps connections to either improve a measure of the geometric quality of the elements (closely related to Q_p^{tet} in (18)) or to maximize the number of elements satisfying the familiar "Delaunay" criterion.

The three primitives, refine, merge, and recon, are combined into the LaGriT grid optimization operation called *massage*. (Massage is similar to the algorithm presented in [10].) Massage accepts three arguments: a creation edge length, an annihilation edge length, and a damage tolerance. (In our numerical runs, we set these three tolerances to .3, .3, and .01, respectively.) Refinement is carried out such that no edge in the grid has length greater than the creation edge length. The refine primitive is invoked using a version of an algorithm due to Rivara [20]. In the algorithm, an edge marked for refinement is placed on a stack. The algorithm then checks that the elements containing the marked edge have no other edges longer than the marked edge. If longer edges are encountered, they are placed on the stack ahead of the marked edge. The process continues recursively. The refinement candidates are then popped off the stack and refined, resulting in a refinement pattern proceeding from the longest edges to the shortest; this pattern is desirable since it usually does not degrade element quality. Massage may attempt to merge pairs of points only if the resulting grid has no edge length greater than the annihilation length. The damage tolerance specifies the amount of change to the shape of a material interface that is permissible. If a node that sits on a material interface is merged out, the interface will become flatter at that point. If the distance from the merged node's original position to its projected position on the flattened interface is greater than the damage tolerance, the merge will not be allowed. Since *merge* and (less commonly) *refine* can produce poorly shaped tetrahedra, then *recon* is used to restore well-shaped elements.

6. Mesh response to grain topology changes: Recoloring.

As grain boundaries move, topology changes must be detected and the mesh must be modified to reflect these topological changes. The topology changes are detected by assembling and monitoring the rate of change of sets of topological components. To detect grain collapse, we assemble sets of connected elements of the same material. For interface surface collapse we assemble sets of connected interface triangles between two materials; for boundary surface detachment, we assemble sets of connected boundary triangles that lie on a given boundary surface; for triple line collapse where a line is surrounded by three or more materials, we assemble sets of connected edges. We monitor the rate of collapse of these sets, and when a collapse or detachment is imminent, the mesh is adjusted. We identify a neighborhood that completely surrounds the collapsing feature and assign a new material to the elements in this neighborhood. We refer to this as *recoloring*. Ideally, the encroaching material that is accumulating most rapidly is chosen to be the new material, but in this first version of our algorithm, the new material is chosen randomly from a list of viable adjacent materials. Soon after the material reassignment, the curvature driven interface motion will effectively straighten the interfaces. Figure 4 is a schematic of three types of topological change. The first frame in each sequence shows the event as it is detected by GRAIN3D; the dotted line demarcates the neighborhood to receive a new material assignment. The second frame shows the mesh just after the material reassignment, and the third frame shows the mesh after the interfaces have straightened. A further topological change, self-pinchoff, where a grain intersects itself and splits into two pieces, is yet to be incorporated in GRAIN3D.

7. Comparison against known solutions. To gauge the accuracy of the GWMFE method, we set up two types of test cases with analytically known collapse rates: spherical collapse and columnar grains.

7.1. Spherically symmetric collapse. The collapse of a spherical grain is easily solvable analytically and provides our first test of the accuracy of GWMFE. Assuming that a sphere is collapsing with normal velocity equal to the curvature (i.e., the *sum* of principle curvatures), we have

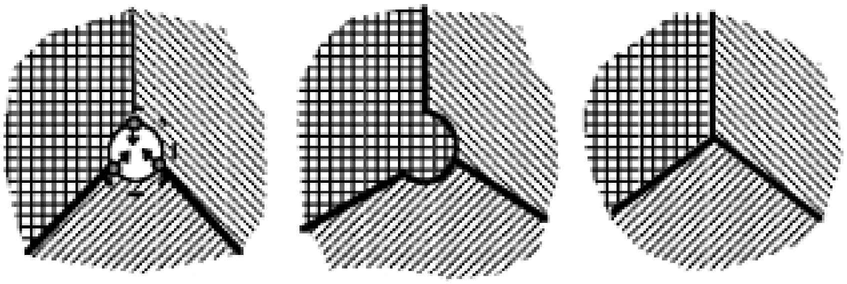
$$\frac{dr}{dt} = K = -\frac{2}{r}.$$

So

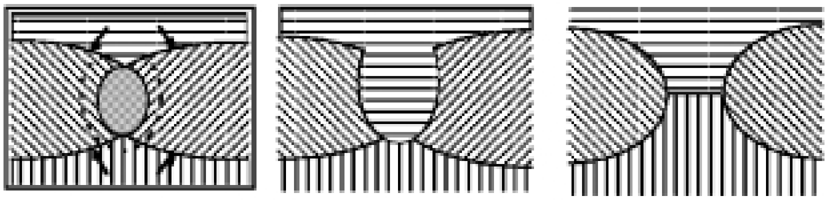
$$\begin{aligned}\frac{dA}{dt} &= \frac{d(4\pi r^2)}{dt}\\ &= 8\pi r\frac{dr}{dt}\\ &= -16\pi.\end{aligned}$$

The rate of change of surface areas is thus constant and is thus a convenient quantity to use for comparison with a numerical GWMFE solution. In Figure 5 we show the initial grid for a 42-node polyhedral representation of the sphere and then at $t = 0.04$. In Figure 6, we do the same for a 162-node sphere. The highest resolution case of a 642-node sphere is not shown. For each case, the triangles on the surface of the sphere are visible. Edges from tetrahedra conformally attached to the outside of the sphere are visible as well. Of course, tetrahedra within the sphere are not visible. Each surface point on the sphere is placed at exactly radius 0.5. We ran the code with $\mu = 1$, $\eta = 10^{-3}$, $h_{\max} = 0.3$, $\epsilon_1 = 5\eta^2$, $\epsilon_2 = 10^{-4}\eta^2\mu$, $\epsilon_3 = \eta\mu/h_{\max}$, $\epsilon_4 = 5\eta^2/h_{\max}$, and $\epsilon_5 = 10^{-4}\eta^2\mu$. In Figures 7–9, we display results for these cases

Collapse of Entire Grain:



Collapse of an Interface between Two Grains:



Collapse of a Triple Line:

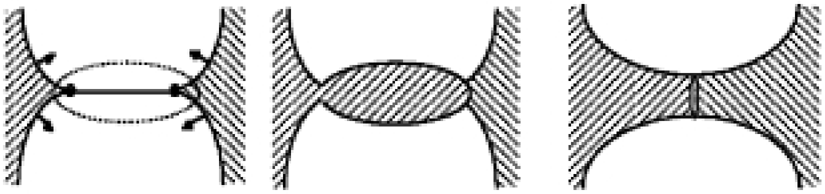


FIG. 4. *Topological collapse detection and resolution by GRAIN3D.*

showing computed graphs of $\frac{dA}{dt}$ vs. t and A vs. t (each dot signifies a time step) and compare them against the analytical results. Note that we apparently converge to the correct $-\frac{dA}{dt}$ as the number of nodes is increased. Also note that at $t = 0$, each numerical solution starts with a surface area less than the exact area π due to the flatness of the triangular facets. As can be seen, this initial surface area deficit is essentially maintained unchanged as the sphere collapses. Hence, errors introduced by the numerical solution seem to be less than errors introduced by simply discretizing the initial condition!

Of course, near the terminal time $t = \frac{\pi}{16\pi} = 0.0625$, we have that the numerical solution for these three cases deviates from the analytic solution. Near the terminal time, the tetrahedra inside the sphere have been collapsed to the point that the shortest altitudes approach η and the tetrahedral quality force (17)–(18) takes over and stops the collapse. If this were a grain growth simulation run, the topology change software would have recolored the collapsing tetrahedra to the color of the surrounding medium close to the terminal time. However, by turning off the recoloring algorithm, we are able to see how the tetrahedral quality force eventually will dominate if no corrective topological change is taken.

To get a better idea of the accuracy of the *pure* GWMFE method without regularization, we reran these three cases with $\mu = 1$, $\eta = 10^{-3}$, $h_{\max} = 0.3$, $\epsilon_1 = .001\eta^2$,

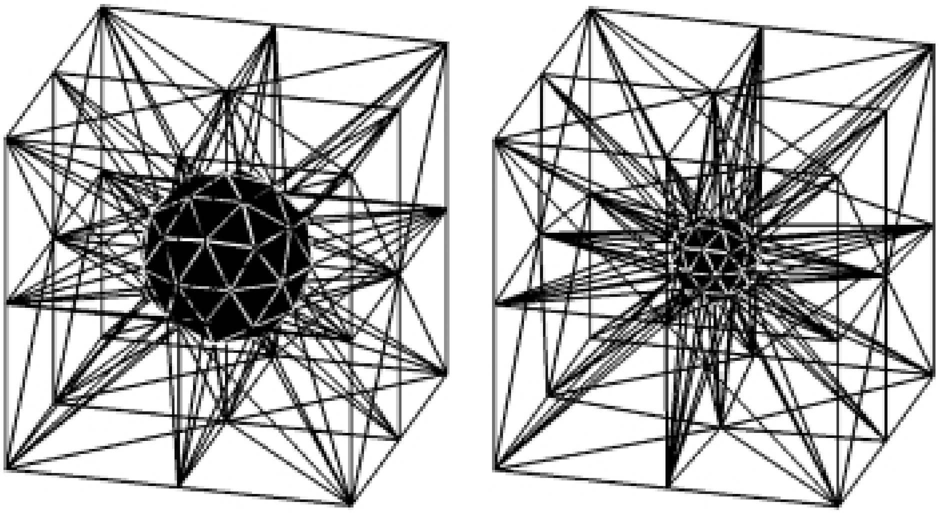


FIG. 5. *Initial* 162*-node sphere (with attached tetrahedra) and at* $t = 0.04$.

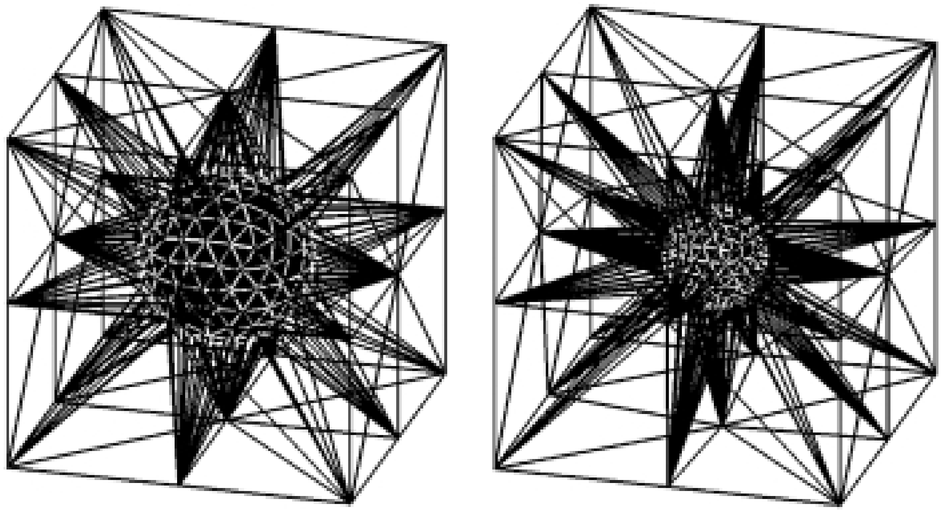


FIG. 6. *Initial* 162*-node sphere (with attached tetrahedra) and at* $t = 0.04$.

$\epsilon_2 = 0$, $\epsilon_3 = 0$, $\epsilon_4 = .001\eta^2 \big/ h_{\max}$, and $\epsilon_5 = 0$, which represent zero right-hand side regularization forces and left-hand side internodal viscosities far smaller than recommended. We plotted $-\frac{dA}{dt}$ for these three cases against the theoretical collapse rate in Figure 10. These simulations abruptly end when the terminal time is reached in the absence of counterbalancing tetrahedral quality forces. It is apparent from this figure that the error in $-\frac{dA}{dt}$ is constant over each run, and that the error is cut by roughly a factor of 4 when one increases the number of nodes from 42 (Case 1) to 162 (Case 2), and then the error drops by a factor of four when the number of nodes is increased again from 162 to 642 (Case 3). Since Case 2 represents a refinement which reduces by a factor of 2 the spacing h between nodes on the initial polyhedron (as compared to the 42-noded polyhedron of Case 1), and since Case 3 represents a further refinement

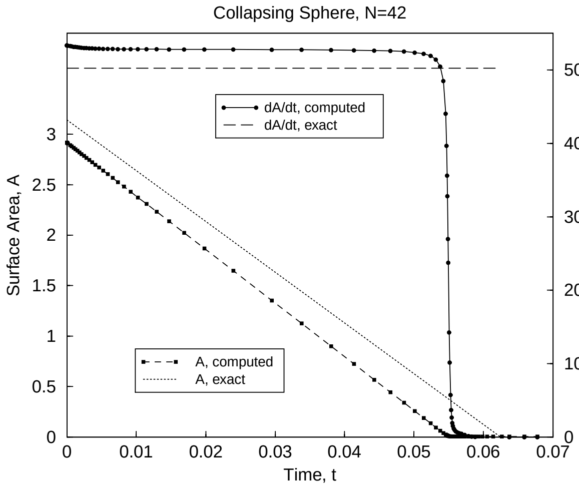


FIG. 7. *Results for* 42*-node sphere run.*

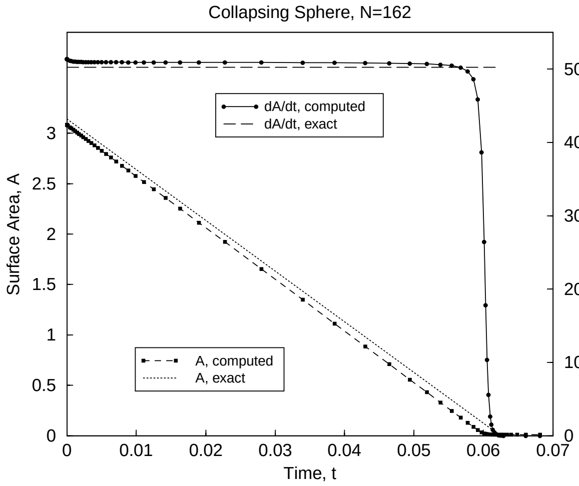


FIG. 8. *Results for* 162*-node sphere run.*

and halving of h, it is tempting to say the error in $-\frac{dA}{dt}$ as computed by the pure GWMFE method is $\mathcal{O}(h^2)$. However, as $t \to 0.0625$, we have that $h \to 0$, but the error remains constant. Instead, consider the Gauss map $\hat{\mathbf{n}} : M \mapsto S^2$ which maps each point $\mathbf{x}$ on a surface M to the unit outward normal $\hat{\mathbf{n}}(\mathbf{x})$ of the surface at $\mathbf{x}$. We define $\Delta\theta$ to be the distance between mapped grid points on the unit sphere. Then at all times for these three numerical test runs, the error in $-\frac{dA}{dt}$ using GWMFE appears to be $\mathcal{O}\big((\Delta\theta)^2\big)$. Thus, it would appear that high accuracy of the GWMFE solution depends on good *angular* resolution of the discretized surface.

The grid maintenance operations in section 5 (which are used in more complex microstructures such as that in section 8) strive to maintain a maximum edge length $h_{\max}$ rather than a coarsest angular resolution $\Delta\theta$. Given the statements of the last paragraph, this implies that small grains in the computations involving grid maintenance operations will have a relatively coarser angular discretization and will suffer

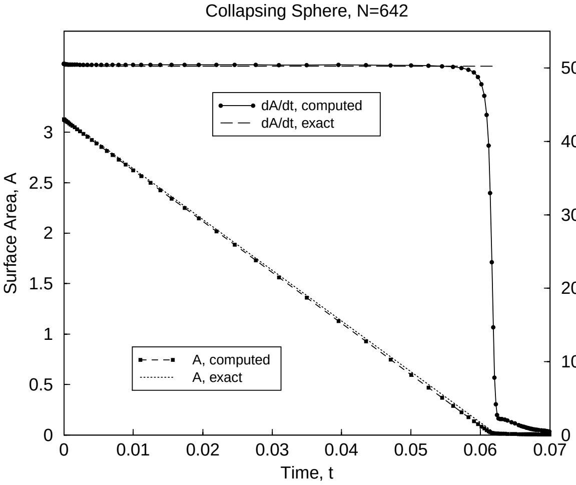


FIG. 9. *Results for* 642*-node sphere run.*

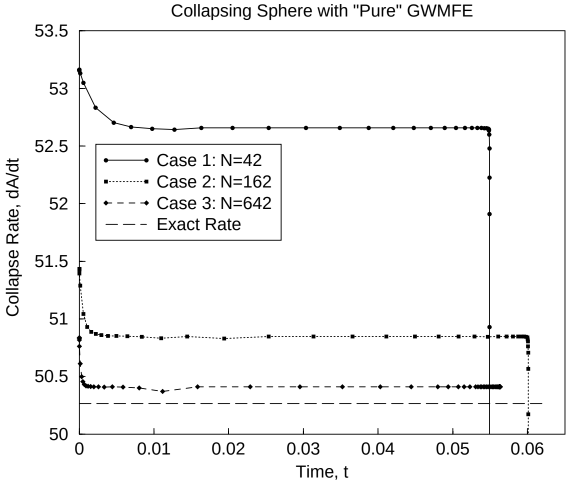


FIG. 10. *Results for nonregularized GWMFE.*

larger computational errors than larger grains. This is somewhat unavoidable, however, since in practical computations grains must be coarsened just prior to collapse.

7.2. Columnar microstructures. Our second set of test cases involve columnar microstructures which are arbitrary 2-D collections of grains that are extended into 3-D to be right cylinders. These microstructures do not possess curvature in 3-D, and thus are amenable to a 2-D analysis.

In 2-D, microstructures moving under curvature driven motion also obey Von Neumann's law [17], which states that for an interior grain G surrounded by n neighbors $G_1, G_2, \ldots, G_n$, the rate of area growth of G is

$$\frac{dA}{dt} = \mu \frac{\pi}{3}\big(n-6\big). \tag{21}$$

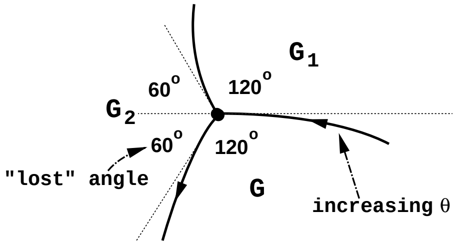


FIG. 11. *Loss of angle at triple point.*

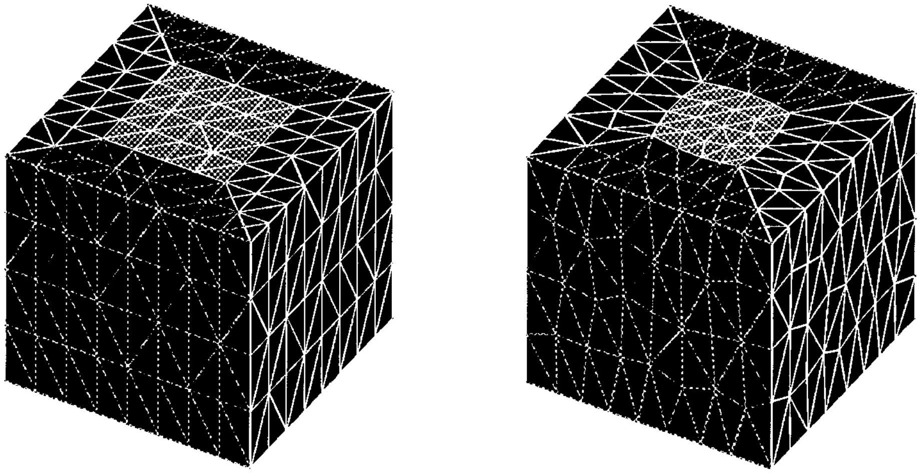


FIG. 12. *Initial* 16*-node columnar microstructure and at* $t = 0.2$*. The term* 16*-node refers to the number of nodes bounding top surface of internal grain.*

This is easily derived as follows. We have that

$$\frac{dA}{dt} = -\int_{\partial G} v_n \, ds,$$

where v_n is normal inward velocity, and the integral is taken over the arclength of the boundary of G. Since $v_n = \mu K$,

$$\begin{aligned}\frac{dA}{dt} &= -\int_{\partial G} \mu K \, ds \\ &= -\mu \int_{\partial G} \frac{d\theta}{ds} \, ds \\ &= -\mu \int_{\partial G} d\theta.\end{aligned}$$

$\int_{\partial G} d\theta$ is not 2π if $n > 0$, since at each triple point junction, there is a 60° loss of angle due to the 120°–120°–120° equilibrium angle condition (Figure 11) that exists at triple points resulting from the necessity of force balance of surface tension at the triple point. The interior grain G has n such triple points. We thus obtain (21).

In Figure 12 we show the initial grid for a five grain columnar microstructure with the axis running top to bottom. A central square-columnar grain G (light in color) is surrounded by four trapezoidal-columnar grains G_1, G_2, G_3, G_4 (darker in color). The area A of the top surface of the central grain is $1 \times 1 = 1$, while the total top area of all five grains put together is $2 \times 2 = 4$. GRAIN3D was run on this initial microstructure using the same parameter values as in the regularized run of section 7.1. Nodes on the exterior planes were allowed to slide on the planes, nodes on exterior edges were allowed to slide on the edges, and the eight corner nodes were fixed. For this run, each of the four interfaces between the four surrounding grains (i.e., $G_1 \cap G_2$, $G_2 \cap G_3$, $G_3 \cap G_4$, and $G_4 \cap G_1$) was associated with and pinned to a vertical exterior edge.

Figure 12 also shows the evolved microstructure, as computed by GRAIN3D, at $t = 0.2$. The grains have remained columnar and the top surface of the central grain has shrunk. The four triple points on the top surface have adopted the correct 120°–120°–120° angle at this time. In Figure 13, we see computed and exact curves for $-\frac{dA}{dt}$ vs. t and A vs. t, where we have used (21) with $n = 4$ to obtain the "exact" value $-\frac{dA}{dt} = \frac{2\pi}{3}$. (We have called this run N = 16, since A is bounded by 16 computational nodes.) For this run, the computed and exact A vs. t are virtually identical. For most of the run, the computed and exact values for $-\frac{dA}{dt}$ are virtually indistinguishable. At the beginning of the run there is a "blip," where $-\frac{dA}{dt}$ is significantly higher than the theoretical value of $\frac{2\pi}{3}$. We attribute this to the state of the four triple points in the initial data. These four triple points are 90°–135°–135° in the initial data, which is not the 120°–120°–120° force balance equilibrium that should exist. Thus the initial blip represents rapid relaxation to the force balance state; essentially the code has accepted "incorrect" initial data (with forces not in balance at the triple points) and relaxed it to an acceptable state on a rapid time scale. The final blip in $-\frac{dA}{dt}$ occurs near the terminal time $t = \frac{3}{2\pi}$ when the central grain has been reduced to an extremely thin column, by which time the recoloring algorithm would have changed the topology, had it been activated. This blip is not surprising since the slender grain carries with it virtually no energy (due to its vanishingly small surface area), and therefore errors present (i.e., due to the nonphysical regularization forces) may well be magnified at this advanced stage.

In Figure 14, we see $-\frac{dA}{dt}$ vs. t and A vs. t for an $N = 32$ geometry (i.e., at twice the grid density of the one pictured in Figure 12). The initial anomalously high $-\frac{dA}{dt}$ blip is shorter lived, and the final blip at the terminal time has disappeared.

A study of "pure" GWMFE error (with regularization turned off) is impossible for this problem. It was possible in the highly symmetrical sphere collapse problem of section 7.1, but in this problem, which exhibits less symmetry and which has changing triple point angles, the regularization forces play a crucial role in preserving the positivity of all triangle areas and tetrahedral volumes throughout the computation.

8. Numerical example with 3-D microstructure. In this section, we show how GRAIN3D evolves a truly 3-D microstructure to steady-state, successfully maintaining mesh quality and performing necessary topological changes during the course of the calculation. In Figure 15(a), we show an initial five grain microstructure provided by the LaGriT tetrahedral mesh generator. We evolve the mesh with GRAIN3D, using the same regularization parameters as in the previous regularization runs. The nodes are allowed to slide on the external surfaces, as in section 7.2. Every $\Delta t_{\text{massage}} = 0.04$ time units, the mesh topology is massaged: edges that are too long are bisected, unneeded nodes are eliminated, and connections are swapped if necessary. Whenever we detect that a topological component is about to collapse (such as a grain, a grain-grain interface, or a triple line) in the next $\Delta t_{\text{collapse}} = 0.002$ time

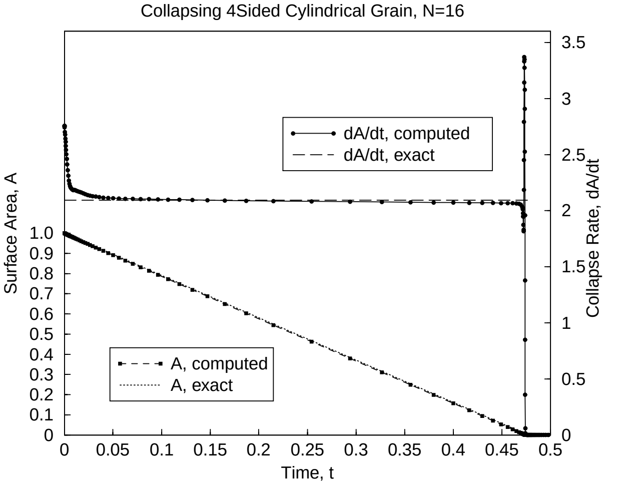


FIG. 13. *Results for* 16*-node run.*

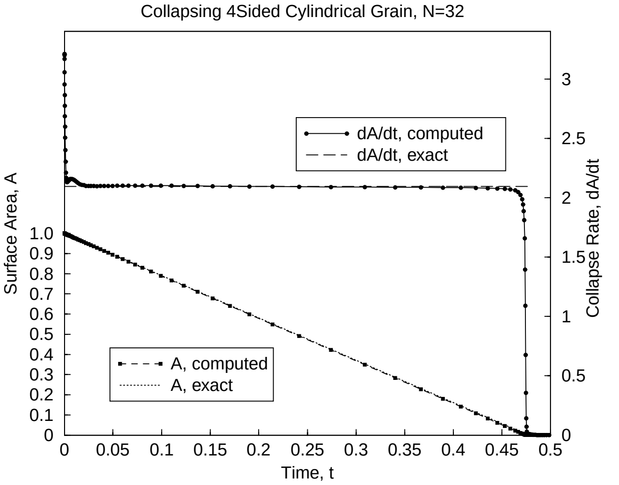
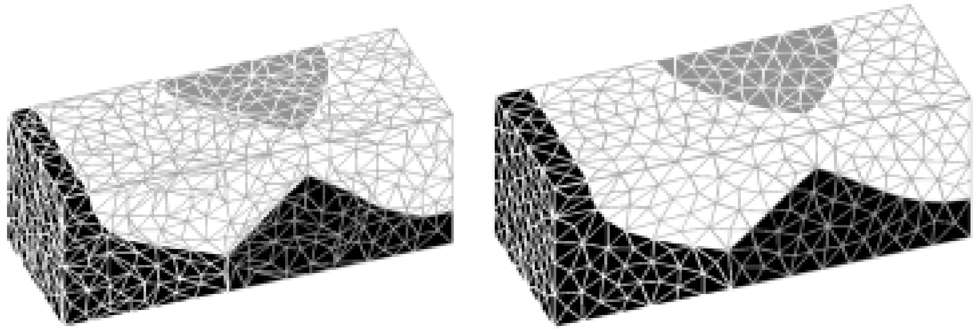


FIG. 14. *Results for* 32*-node run.*

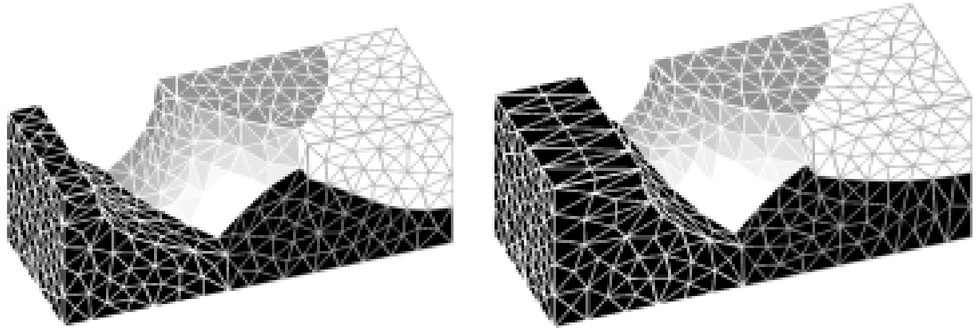
units, we call the recoloring algorithm to effectively perform the necessary topological change.

The massage algorithm is called immediately after $t = 0.00$, and the regularization forces (especially perimeter dependent surface energy (12)) fatten the elements, so by $t = 0.02$ (Figure 15(b)), the mesh is much improved. In 15(c), we have made one grain graphically transparent to reveal triples lines and quadruple points. At $t = 0.20$ (15(d)), the invisible grain has shrunk. At $t = 0.30$ (15(e)), the invisible grain has detached from the rear wall. In 15(f), the invisible grain is near collapse. By 15(g), the grain has been eliminated by the recoloring algorithm. This figure also shows a small rear grain about to collapse. In 15(i) this grain has collapsed and we are down to three grains. In fact, the microstructure is now columnar (the axis runs front to back) and would be perfectly well modeled by a 2-D front tracking code, such as that described in [5]. At $t = 1.50$ (15(j)), the light colored grain has disappeared, leaving



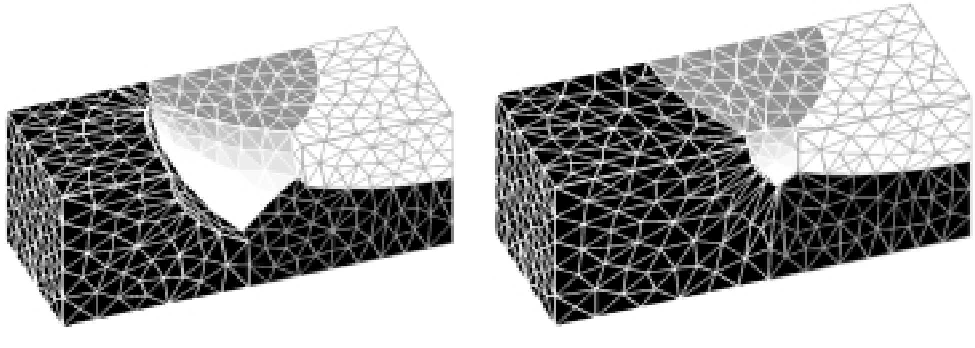
(a) $t = 0.00$. Initial grid is from LaGriT.

(b) $t = 0.02$. Grid improved by regularization and massage.



(c) $t = 0.02$. Grid with grain graphically removed for illustration.

(d) $t = 0.02$. Invisible grain has shrunk.

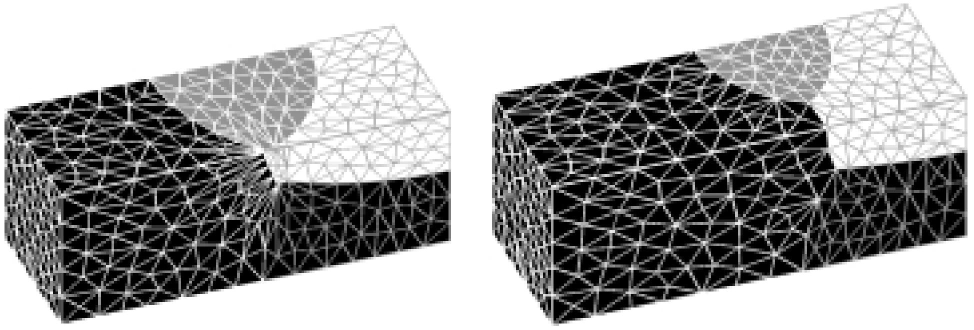


(e) $t = 0.30$. Invisible grain has detached from rear wall.

(f) $t = 0.45$. Invisible grain about to collapse.

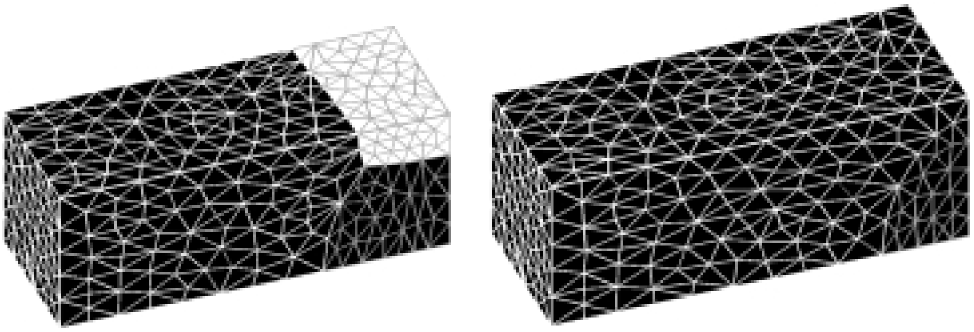
FIG. 15 (a)–(f). *Grain evolution time sequence computed by GRAIN3D.*

a collapsing circular-columnar grain and a larger complementary grain. At $t = 1.75$ (15(k)), the circular-columnar grain has collapsed, leaving the entire computational domain covered by a single grain. Evidence of the grain collapse is visible in that the grid is somewhat finer in the bottom right-hand corner. Finally, at $t = 1.80$ (15(l)), massage and regularization have coarsened the grid in the vicinity of the last collapsed grain. Of course, the difference between the grids at $t = 1.75$ and $t = 1.80$ is



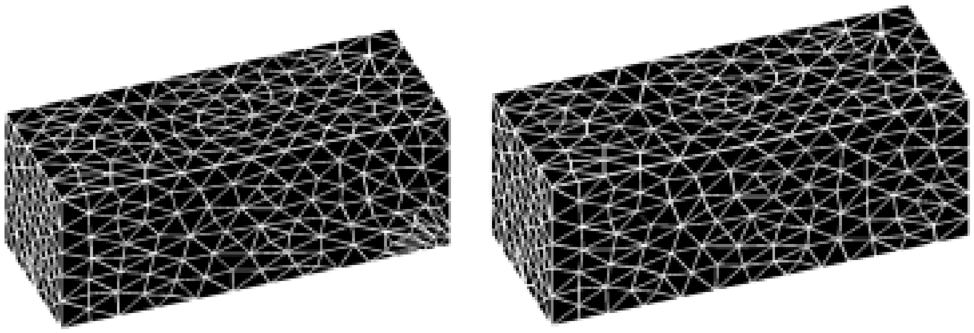
(g) $t = 0.47$. Invisible grain has collapsed.

(h) $t = 0.80$. Surface triple points draw apart.



(i) $t = 1.10$. A second grain has collapsed. The microstructure is now 2-D.

(j) $t = 1.50$. A third grain has collapsed.



(k) $t = 1.75$. Fourth grain has collapsed. One grain left.

(l) $t = 1.80$. Grid improved by regularization and massage.

FIG. 15 (g)–(l). *Grain evolution time sequence computed by GRAIN3D.*

purely nonphysical—the grain is the same, but the grid is different. The run is truly completed at $t = 1.75$ when the microstructure evolution has ceased, but it isn't until $t = 1.80$ that the grid has settled down.

Acknowledgments. The author would like to acknowledge the invaluable support of the rest of the Los Alamos T-1 Grain Growth team: Denise George helped with enhancements to LaGriT, Tinka Gammel generated the initial microstructure used in section 8, Neil Carlson (now with Motorola, Inc.) provided assistance with his

GWMFE2DS code, and Galen Straub provided necessary motivation and resources. I also thank the referees for their detailed comments and corrections.

REFERENCES

[1] M.J. Baines, *Moving Finite Elements,* Oxford University Press, London, 1994.

[2] K. Brakke, *The surface evolver,* Experiment. Math., 1 (1992), pp. 141–165.

[3] N.N. Carlson and K. Miller, *Design and application of a gradient-weighted moving finite element code* I: *In one dimension,* SIAM J. Sci. Comput., 19 (1998), pp. 728–765.

[4] N.N. Carlson and K. Miller, *Design and application of a gradient-weighted moving finite element code* II: *In two dimensions,* SIAM J. Sci. Comput., 19 (1998), pp. 766–798.

[5] H.J. Frost, C.V. Thompson, C.L. Howe, and J. Whang, *A two-dimensional computer simulation of capillarity-driven grain growth: Preliminary results,* Scripta Met., 22 (1988), pp. 65–70.

[6] K. Fuchizaki and K. Kawasaki, 3*-dimensional computer modeling of grain-growth: A vertex model approach,* Materials Science Forum, 204 (1996), pp. 267–278.

[7] J.T. Gammel and A. Kuprat, *Modeling Metallic Microstructure: Incorporating Grain Boundary Orientation Dependence,* Tech. report LA-UR-98-1150, in the Special Features Supplement to the Theoretical Division Self-Assessment, Los Alamos National Laboratory, Los Alamos, NM, Spring 1998.

[8] D.C. George, *LaGriT User's Manual,* http://www.t12.lanl.gov/~lagrit.

[9] K. Kawasaki, T. Nagai, and K. Nakashima, *Vertex models for two-dimensional grain growth,* Phil. Mag. B, 60 (1989), p. 399.

[10] A. Kuprat, *Creation and Annihilation of Nodes for the Moving Finite Element Method,* Ph.D. thesis, University of California, Berkeley, May 1992.

[11] A. Liu and B. Joe, *Relationship between tetrahedron shape measures,* BIT, 34 (1994), pp. 268–287.

[12] P. Marcellini and K. Miller, *Regularization for prescribed mean curvature and for motion by mean curvature,* in Curvature Flows and Related Topics, Proceedings of the Second International Conference at Levico, Italy, 1994, A. Damlamian, J. Spruck, and A. Visintin, eds., Gakkotosho, Tokyo, 1995, pp. 145–158.

[13] P. Marcellini and K. Miller, *Elliptic versus parabolic regularization for the equation of prescribed mean curvature,* J. Differential Equations, 137 (1997), pp. 1–53.

[14] K. Miller, *Moving finite elements.* II, SIAM J. Numer. Anal., 18 (1981), pp. 1033–1057.

[15] K. Miller, *A geometrical-mechanical interpretation of gradient-weighted moving finite elements,* SIAM J. Numer. Anal., 34 (1997), pp. 67–90.

[16] K. Miller and R.N. Miller, *Moving finite elements.* I, SIAM J. Numer. Anal., 18 (1981), pp. 1019–1032.

[17] W.W. Mullins, *Two-dimensional motion of idealized grain boundaries,* J. Appl. Phys., 27 (1956), pp. 900–904.

[18] S. Nazari, *Rotational Surfaces in Euclidean and Hyperbolic Spaces, Mean Curvature Motion, and the Moving Finite Element Method,* Ph.D. thesis, Department of Mathematics, University of California, Berkeley, 1993.

[19] D.A. Porter and K.E. Easterling, *Phase Transformations in Metals and Alloys,* 2nd ed., Chapman and Hall, London, 1992.

[20] M. Rivara, *New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations,* Internat. J. Numer. Methods Engrg., 40 (1997), pp. 3313–3324.

[21] S.J. Ruuth, *A diffusion-generated approach to multiphase motion,* J. Comput. Phys., 145 (1998), pp. 166–192.

[22] Y. Saad and M.H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[23] J.A. Sethian, *Level Set Methods,* Cambridge University Press, London, 1996.

[24] E.Z. Xaba, *Robust Iterative Solvers for Linear and Nonlinear Finite Element Equations,* Ph.D. thesis, Department of Mathematics, University of California, Berkeley, 1997.

[25] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, *A variational level set approach to multiphase motion,* J. Comput. Phys., 127 (1996), pp. 179–195.